

CLAIMS

What is claimed is:

5

1. A method for processing objects at a data processing system in a network, the method comprising:

receiving a first message at a computing device; and

determining that a message header in the first

10 message indicates that the first message relates to a fragment.

2. The method of claim 1 further comprising:

storing a fragment from the first message in a cache

15 maintained by a cache management unit within the

computing device, wherein the cache management unit

operates equivalently in support of fragment caching

operations whether the computing device acts as a client,

a server, or a hub located throughout the network.

20

3. The method of claim 1 further comprising:

determining that a message header in the first

message indicates that a message body portion of the

first message is a fragment.

25

4. The method of claim 1 further comprising:

determining that a message header in the first

message indicates that the fragment is cacheable.

1003477 "E" 121901

5. The method of claim 4 wherein the first message comprises an indication that the fragment is non-cacheable to non-fragment-supporting cache management units and an indication that the fragment is cacheable to fragment-supporting cache management units.

6. The method of claim 5 wherein the first message comprises an HTTP Cache-Control header with a no-cache directive for non-fragment-supporting cache management units and with a directive for caching the fragment for fragment-supporting cache management units.

7. The method of claim 1 further comprising:
receiving a second message at the computing device,
wherein the second message comprises a request for the fragment;
searching the cache in response to receiving the second message;
retrieving the fragment from the cache in response to searching the cache; and
sending the fragment in a third message to an originator of the second message without sending the second message to its destination address.

8. The method of claim 7 wherein the second message comprises information indicating that a page assembly operation is not required at the computing device prior to returning the third message.

9. The method of claim 7 wherein the second message comprises a message header with a directive indicating that the third message will be received by a second computing device that has a fragment-supporting cache management unit.

10. The method of claim 7 wherein the second message does not comprise a message header with a directive indicating that the third message will be received by a second computing device that has a fragment-supporting cache management unit.

11. The method of claim 7 further comprising:
performing a page assembly operation at the computing device prior to sending the third message.

12. The method of claim 1 further comprising:
performing a page assembly operation at the computing device to form an assembled fragment.

13. The method of claim 12 further comprising:
determining whether the fragment is a top-level fragment that includes a link to a next-level fragment;
retrieving the next-level fragment in response to determining that the fragment is a top-level fragment that includes a link to a next-level fragment; and
combining the top-level fragment and the next-level fragment into an assembled fragment.

14. The method of claim 13 further comprising:
embedding content of the next-level fragment into
content of the top-level fragment.

5 15. The method of claim 13 further comprising:
generating a property value for the assembled
fragment from a property value of the top-level fragment
and a property value of the next-level fragment.

10 16. The method of claim 13 further comprising:
computing a header value or directive for the
assembled fragment from a header value or directive of
the top-level fragment and a header value or directive of
the next-level fragment.

15 17. The method of claim 12 further comprising:
generating a fourth message comprising the assembled
fragment, wherein the fourth message is an HTTP
(Hypertext Transport Protocol) Response message.

20 18. The method of claim 17 further comprising:
determining a shortest expiration time for the
top-level fragment and the next-level fragment; and
setting an "Expires" header in the fourth message to
25 the shortest expiration time.

19. The method of claim 17 further comprising:
determining a smallest maximum age for the top-level
fragment and the next-level fragment; and
30 setting a "Cache-Control: max-age" directive in the
fourth message to the smallest maximum age.

FOOTNOTES

10

15

20

25

24. The method of claim 1 further comprising:

retrieving a set of fragment caching rules from the first message, wherein a fragment caching rule determines a manner for generating a cache identifier for the

5 fragment; and

generating a cache identifier for the fragment in accordance with a fragment caching rule.

25. The method of claim 24 further comprising:

10 uniquely identifying the fragment using the cache identifier.

26. The method of claim 24 further comprising:

15 performing the storing operation using the generated cache identifier for the fragment.

27. The method of claim 24 further comprising:

20 obtaining at least a path portion of a URI (Uniform Resource Identifier) associated with the fragment in order to form a base cache identifier; and

applying a fragment caching rule to the base cache identifier to form a cache identifier for the fragment, wherein a fragment caching rule comprises a set of query parameter names and/or cookie names that are used to
25 obtain name-value pairs that are appended to the base cache identifier.

100447 "SECRET"

```
5    fragment;
```

10 fragment.

29. The method of claim 28 further comprising:

```
15 message is a fragment.
```

inserting in the response message a message header that indicates that the fragment is cacheable.

20

25 indication that the fragment is cacheable to

32. An apparatus for processing objects at a data processing system in a network, the apparatus comprising:

means for receiving a first message at a computing device; and

5 means for determining that a message header in the first message indicates that the first message relates to a fragment.

33. The apparatus of claim 32 further comprising:

10 means for storing a fragment from the first message in a cache maintained by a cache management unit within the computing device, wherein the cache management unit operates equivalently in support of fragment caching operations whether the computing device acts as a client, 15 a server, or a hub located throughout the network.

34. The apparatus of claim 32 further comprising:

20 means for determining that a message header in the first message indicates that a message body portion of the first message is a fragment.

35. The apparatus of claim 32 further comprising:

25 means for determining that a message header in the first message indicates that the fragment is cacheable.

36. The apparatus of claim 35 wherein the first message comprises an indication that the fragment is non-cacheable to non-fragment-supporting cache management units and an indication that the fragment is cacheable to 30 fragment-supporting cache management units.

37. The apparatus of claim 36 wherein the first message comprises an HTTP Cache-Control header with a no-cache directive for non-fragment-supporting cache management units and with a directive for caching the fragment for
5 fragment-supporting cache management units.

38. The apparatus of claim 32 further comprising:
means for receiving a second message at the
computing device, wherein the second message comprises a
10 request for the fragment;
means for searching the cache in response to
receiving the second message;
means for retrieving the fragment from the cache in
response to searching the cache; and
15 means for sending the fragment in a third message to
an originator of the second message without sending the
second message to its destination address.

39. The apparatus of claim 38 wherein the second message
20 comprises information indicating that a page assembly
operation is not required at the computing device prior
to returning the third message.

40. The apparatus of claim 38 wherein the second message
25 comprises a message header with a directive indicating
that the third message will be received by a second
computing device that has a fragment-supporting cache
management unit.

41. The apparatus of claim 38 wherein the second message does not comprise a message header with a directive indicating that the third message will be received by a second computing device that has a fragment-supporting cache management unit.

42. The apparatus of claim 38 further comprising:
means for performing a page assembly operation at the computing device prior to sending the third message.

43. The apparatus of claim 32 further comprising:
means for performing a page assembly operation at the computing device to form an assembled fragment.

44. The apparatus of claim 43 further comprising:
means for determining whether the fragment is a top-level fragment that includes a link to a next-level fragment;
means for retrieving the next-level fragment in response to determining that the fragment is a top-level fragment that includes a link to a next-level fragment;
and
means for combining the top-level fragment and the next-level fragment into an assembled fragment.

45. The apparatus of claim 44 further comprising:
means for embedding content of the next-level fragment into content of the top-level fragment.

46. The apparatus of claim 44 further comprising:
means for generating a property value for the
assembled fragment from a property value of the top-level
fragment and a property value of the next-level fragment.

5

47. The apparatus of claim 44 further comprising:
means for computing a header value or directive for
the assembled fragment from a header value or directive
of the top-level fragment and a header value or directive
of the next-level fragment.

10

48. The apparatus of claim 43 further comprising:
means for generating a fourth message comprising the
assembled fragment, wherein the fourth message is an HTTP
(Hypertext Transport Protocol) Response message.

15

49. The apparatus of claim 48 further comprising:
means for determining a shortest expiration time for
the top-level fragment and the next-level fragment; and
means for setting an "Expires" header in the fourth
message to the shortest expiration time.

20

50. The apparatus of claim 48 further comprising:
means for determining a smallest maximum age for the
top-level fragment and the next-level fragment; and
means for setting a "Cache-Control: max-age"
directive in the fourth message to the smallest maximum
age.

25

TOP SECRET

51. The apparatus of claim 48 further comprising:

means for computing a sum of content length values
for the top-level fragment and the next-level fragment;
and

5 means for setting a "Content-Length" header in the
fourth message to the sum of the content length values.

52. The apparatus of claim 48 further comprising:

10 means for determining a latest modification time for
the top-level fragment and the next-level fragment; and
means for setting a "Last-Modified" header in the
fourth message to the latest modification time.

53. The apparatus of claim 32 further comprising:

15 means for retrieving a set of dependency identifiers
from the first message, wherein a dependency identifier
is generated by a server that originated the fragment;
and

20 means for storing the set of dependency identifiers
in association with a source identifier for the fragment.

54. The apparatus of claim 53 further comprising:

25 means for receiving an invalidation request message;
means for retrieving a dependency identifier from
the invalidation request message;

means for determining a set of fragments that are
associated with the dependency identifier; and

30 means for purging the set of fragments from the
cache in response to determining the set of fragments
that are associated with the dependency identifier.

1004477e 121501

55. The apparatus of claim 32 further comprising:

means for retrieving a set of fragment caching rules from the first message, wherein a fragment caching rule determines a manner for generating a cache identifier for the fragment; and

means for generating a cache identifier for the fragment in accordance with a fragment caching rule.

56. The apparatus of claim 55 further comprising:

means for uniquely identifying the fragment using the cache identifier.

57. The apparatus of claim 55 further comprising:

means for performing the storing operation using the generated cache identifier for the fragment.

58. The apparatus of claim 55 further comprising:

means for obtaining at least a path portion of a URI (Uniform Resource Identifier) associated with the fragment in order to form a base cache identifier; and

means for applying a fragment caching rule to the base cache identifier to form a cache identifier for the fragment, wherein a fragment caching rule comprises a set of query parameter names and/or cookie names that are used to obtain name-value pairs that are appended to the base cache identifier.

100347-100347

59. An apparatus for processing objects at a data processing system in a network, the apparatus comprising:

means for receiving a request message at a server,
wherein the request message comprises a source identifier
5 for a fragment;

means for generating a response message comprising
the fragment; and

means for inserting in the response message a
message header that indicates that the first message
10 relates to a fragment.

60. The apparatus of claim 59 further comprising:

means for inserting in the response message a
message header that indicates a message body portion of
15 the response message is a fragment.

61. The apparatus of claim 59 further comprising:

means for inserting in the response message a
message header that indicates that the fragment is
20 cacheable.

62. The apparatus of claim 61 further comprising:

means for inserting in the response message a
message header that indicates that the fragment is
25 non-cacheable to non-fragment-supporting cache management
units and an indication that the fragment is cacheable to
fragment-supporting cache management units.

10034724001

63. A computer program product in a computer readable medium for use in a data processing system in a network for processing objects, the computer program product comprising:

5 instructions for receiving a first message at a computing device; and

 instructions for determining that a message header in the first message indicates that the first message relates to a fragment.

10

64. The computer program product of claim 63 further comprising:

 instructions for storing a fragment from the first message in a cache maintained by a cache management unit within the computing device, wherein the cache management unit operates equivalently in support of fragment caching operations whether the computing device acts as a client, a server, or a hub located throughout the network.

15
20 65. The computer program product of claim 63 further comprising:

 instructions for determining that a message header in the first message indicates that a message body portion of the first message is a fragment.

25

66. The computer program product of claim 63 further comprising:

 instructions for determining that a message header in the first message indicates that the fragment is
30 cacheable.

1003477 "E" 1991

67. The computer program product of claim 66 wherein the first message comprises an indication that the fragment is non-cacheable to non-fragment-supporting cache management units and an indication that the fragment is cacheable to fragment-supporting cache management units.

68. The computer program product of claim 67 wherein the first message comprises an HTTP Cache-Control header with a no-cache directive for non-fragment-supporting cache management units and with a directive for caching the fragment for fragment-supporting cache management units.

69. The computer program product of claim 63 further comprising:

- instructions for receiving a second message at the computing device, wherein the second message comprises a request for the fragment;
- instructions for searching the cache in response to receiving the second message;
- instructions for retrieving the fragment from the cache in response to searching the cache; and
- instructions for sending the fragment in a third message to an originator of the second message without sending the second message to its destination address.

70. The computer program product of claim 69 wherein the second message comprises information indicating that a page assembly operation is not required at the computing device prior to returning the third message.

71. The computer program product of claim 69 wherein the second message comprises a message header with a directive indicating that the third message will be received by a second computing device that has a fragment-supporting cache management unit.

72. The computer program product of claim 69 wherein the second message does not comprise a message header with a directive indicating that the third message will be received by a second computing device that has a fragment-supporting cache management unit.

73. The computer program product of claim 69 further comprising:
instructions for performing a page assembly operation at the computing device prior to sending the third message.

74. The computer program product of claim 63 further comprising:
instructions for performing a page assembly operation at the computing device to form an assembled fragment.

75. The computer program product of claim 74 further comprising:

instructions for determining whether the fragment is a top-level fragment that includes a link to a next-level fragment;

instructions for retrieving the next-level fragment in response to determining that the fragment is a top-level fragment that includes a link to a next-level fragment; and

instructions for combining the top-level fragment and the next-level fragment into an assembled fragment.

76. The computer program product of claim 75 further comprising:

instructions for embedding content of the next-level fragment into content of the top-level fragment.

77. The computer program product of claim 75 further comprising:

instructions for generating a property value for the assembled fragment from a property value of the top-level fragment and a property value of the next-level fragment.

78. The computer program product of claim 75 further comprising:

instructions for computing a header value or directive for the assembled fragment from a header value or directive of the top-level fragment and a header value or directive of the next-level fragment.

TOP SECRET

79. The computer program product of claim 74 further comprising:

instructions for generating a fourth message comprising the assembled fragment, wherein the fourth
5 message is an HTTP (Hypertext Transport Protocol) Response message.

80. The computer program product of claim 79 further comprising:

10 instructions for determining a shortest expiration time for the top-level fragment and the next-level fragment; and

instructions for setting an "Expires" header in the fourth message to the shortest expiration time.

15 81. The computer program product of claim 79 further comprising:

instructions for determining a smallest maximum age for the top-level fragment and the next-level fragment;
20 and

instructions for setting a "Cache-Control: max-age" directive in the fourth message to the smallest maximum age.

400347-1504

82. The computer program product of claim 79 further comprising:

instructions for computing a sum of content length values for the top-level fragment and the next-level

5 fragment; and

instructions for setting a "Content-Length" header in the fourth message to the sum of the content length values.

10 83. The computer program product of claim 79 further comprising:

instructions for determining a latest modification time for the top-level fragment and the next-level fragment; and

15 instructions for setting a "Last-Modified" header in the fourth message to the latest modification time.

84. The computer program product of claim 63 further comprising:

20 instructions for retrieving a set of dependency identifiers from the first message, wherein a dependency identifier is generated by a server that originated the fragment; and

25 instructions for storing the set of dependency identifiers in association with a source identifier for the fragment.

TOP SECRET

85. The computer program product of claim 84 further comprising:

instructions for receiving an invalidation request message;

5 instructions for retrieving a dependency identifier from the invalidation request message;

instructions for determining a set of fragments that are associated with the dependency identifier; and

10 instructions for purging the set of fragments from the cache in response to determining the set of fragments that are associated with the dependency identifier.

86. The computer program product of claim 63 further comprising:

15 instructions for retrieving a set of fragment caching rules from the first message, wherein a fragment caching rule determines a manner for generating a cache identifier for the fragment; and

20 instructions for generating a cache identifier for the fragment in accordance with a fragment caching rule.

87. The computer program product of claim 86 further comprising:

25 instructions for uniquely identifying the fragment using the cache identifier.

88. The computer program product of claim 86 further comprising:

30 instructions for performing the storing operation using the generated cache identifier for the fragment.

FOR OFFICIAL USE ONLY

89. The computer program product of claim 86 further comprising:

instructions for obtaining at least a path portion
of a URI (Uniform Resource Identifier) associated with
5 the fragment in order to form a base cache identifier;
and

instructions for applying a fragment caching rule to
the base cache identifier to form a cache identifier for
the fragment, wherein a fragment caching rule comprises a
10 set of query parameter names and/or cookie names that are
used to obtain name-value pairs that are appended to the
base cache identifier.

100347 " 121901
105121 " 224500

90. A computer program product in a computer readable medium for use in a data processing system in a network for processing objects, the computer program product comprising:

- 5 instructions for receiving a request message at a server, wherein the request message comprises a source identifier for a fragment;
- instructions for generating a response message comprising the fragment; and
- 10 instructions for inserting in the response message a message header that indicates that the first message relates to a fragment.

91. The computer program product of claim 90 further comprising:
- 15 instructions for inserting in the response message a message header that indicates a message body portion of the response message is a fragment.

- 20 92. The computer program product of claim 90 further comprising:
- instructions for inserting in the response message a message header that indicates that the fragment is cacheable.

93. The computer program product of claim 92 further comprising:

- instructions for inserting in the response message a message header that indicates that the fragment is
- 5 non-cacheable to non-fragment-supporting cache management units and an indication that the fragment is cacheable to fragment-supporting cache management units.

FOR OFFICIAL USE ONLY

94. A data structure for use by a computing device in defining a message that is transmitted on a network, the data structure comprising:

an indicator that the message is a request message
5 or a response message; and

a fragment header comprising a keyword that indicates that the message is to be processed by a fragment-supporting cache management unit and one or more fragment header directives that indicate the manner in
10 which the message is to be processed.

95. The data structure of claim 94 wherein a fragment-supporting cache management unit is located within the computing device and operates equivalently in
15 support of fragment caching operations without regard to whether the computing device acts as a client, a server, or a hub located throughout the network.

96. The data structure of claim 94 further comprising:
20 a fragment header directive for inclusion in a request message to indicate that a computing device that processed the request message has a fragment-supporting cache management unit.

97. The data structure of claim 94 further comprising:
25 a fragment header directive for inclusion in a response message to indicate a set of dependency identifiers that are used by an origin server to purge a fragment within the response message from a cache
30 maintained by a fragment-supporting cache management unit.

1007-2-24-001

98. The data structure of claim 94 further comprising:

5 a fragment header directive for inclusion in a
response message to indicate a set of fragment caching
rules that are used to form a cache identifier that
uniquely identifies a fragment in the response message.

99. The data structure of claim 94 wherein the request
message or response message is an HTTP (Hypertext

10 Transport Protocol) Request message or an HTTP Response
message.

1003472 "111601
1003472 "111601

100. A data structure for use by a computing device in defining a content object, the data structure comprising:

- a set of delimiters for a markup language element;
- a keyword for indicating that the markup language
- 5 element is a link to a fragment; and
- a source identifier for the fragment, wherein the source identifier is used by a fragment-supporting cache management unit to retrieve the fragment.

- 10 101. The data structure of claim 100 wherein a fragment-supporting cache management unit is located within the computing device and operates equivalently in support of fragment caching operations without regard to whether the computing device acts as a client, a server,
- 15 or a hub located throughout the network.

102. The data structure of claim 100 further comprising:
- an alternate source identifier for the fragment,
- wherein the alternate source identifier can be used by
- 20 the fragment-supporting cache management unit to retrieve the fragment.

103. The data structure of claim 100 further comprising:
- a set of query parameters that are appended to the
- 25 source identifier.

104. The data structure of claim 100 wherein the markup language is SGML (Standard Generalized Markup Language).

105. The data structure of claim 104 wherein the markup language element is compatible with HTML (Hypertext Markup Language).

FIG. 10 is a schematic diagram of the data structure of claim 104.